

IMPLEMENTAÇÃO EM FPGA DE UM ESTIMADOR DE FASOR EM PRESENÇA DE COMPONENTE DE DECAIMENTO CC EXPONENCIAL

Leandro Rodrigues Manso Silva*
Rodrigo Vianello**
Carlos Augusto Duque***

RESUMO

Este trabalho apresenta a implementação em “Field Programmable Gate Array” (FPGA) de um algoritmo de estimação de fasores na presença de decaimento exponencial (decaimento CC). O método proposto é baseado na estimação dos parâmetros do decaimento exponencial e faz uso de técnicas de processamento de sinais tais como janelamento, “Discrete Time Fourier Transform” (DTFT) e busca em tabelas. O método é capaz de estimar a componente fundamental em aproximadamente meio ciclo da componente fundamental. Resultados de simulação em tempo real, usando aritmética de ponto fixo, são comparados com os gerados em ponto flutuante e “off-line” a partir do MATLAB. Casos exemplos sintéticos e reais foram utilizados para validar o algoritmo de estimação.

A estimação do fasor da componente fundamental é realizada pela aplicação de filtros discretos de Fourier (DFT). Paralelamente a esse processo é feito o cálculo das correções devido à presença da componente exponencial, sendo que o resultado final é obtido quando as correções são aplicadas ao algoritmo de estimação.

Para a obtenção das correções, o sinal é analisado utilizando uma janela adequada que contém zeros em locais estratégicos do espectro. Após o janelamento é calculada a DTFT do sinal resultante na frequência de 53 Hz, visto que foi a frequência que apresentou uma melhor curva para a criação (discretização) da curva armazenada na tabela k_1 . A partir desses resultados estimam-se os parâmetros do decaimento CC, através de busca em tabelas. As tabelas são geradas off-line e armazenadas em memórias ROM instanciadas no código.

A fim de simplificar a implementação foram utilizadas algumas “Megacores” disponibilizadas pelo fabricante (Altera®), tais como, memórias ROM e divisores. Assim o foco do trabalho foi na síntese dos filtros e do algoritmo de busca em tabelas. Apesar de estar em nível de simulação, o código foi escrito em linguagem sintetizável, ou seja, está pronto para ser gravado no chip.

Palavras-chave: Estimação, Decaimento CC exponencial, Janelamento, FPGA, Busca em tabelas, Filtros, DFT, DTFT, Correções.

* Graduado e Mestrando em Engenharia Elétrica na UFJF e Bolsista de Iniciação Científica (PIBIC/CNPq)

** Mestre em Engenharia Elétrica pela UFJF

*** Professor Orientador do Departamento de Circuitos Elétricos – UFJF - carlos.duque@ufjf.edu.br; Departamento de Circuitos Elétricos, Faculdade de Engenharia, UFJF, Cidade Universitária, Bairro Martelos, CEP 36036-330, Juiz de Fora, MG.

1 Introdução

O presente trabalho apresenta a implementação em “Field Programmable Gate Array” (FPGA) de um algoritmo de estimação de fasores na presença de decaimento CC exponencial. O método é baseado na estimação dos parâmetros do decaimento exponencial e faz uso de técnicas de processamento de sinais tais como janelamento, “Discrete Time Fourier Transform” (DTFT) e busca em tabelas. O método é capaz de estimar a componente fundamental em aproximadamente meio ciclo da componente fundamental.

As interconexões do sistema elétrico de potência podem se tornar altamente instáveis quando uma falta não é eliminada do sistema rapidamente (SIDHU et al., 2005). Caso uma falta venha a acontecer em uma linha de transmissão, esta deve ser identificada e eliminada no menor tempo possível, a fim de evitar danos às partes vitais do sistema. Para isso, os algoritmos dos relés devem identificar a falta o mais rápido possível, tipicamente em um ciclo, ou menos, da componente de frequência fundamental do sistema, o que torna o processo de estimação um desafio, uma vez que o trinômio velocidade, complexidade e precisão precisa ser garantido.

Na maior parte dos algoritmos implementados nos relés numéricos, um filtro digital extrai a componente fundamental do sinal. Este processo deve ser rápido e preciso. Normalmente os sinais advindos das faltas são compostos pela componente de frequência fundamental, harmônicos e uma componente de decaimento CC exponencial. Os filtros digitais então devem extrair somente a componente de frequência fundamental, livre do decaimento CC, rejeitando as demais componentes.

A constante de tempo e a amplitude do decaimento CC exponencial são desconhecidas e associadas com a resistência de falta, a posição da falta e o momento inicial da falta. A constante de tempo é geralmente determinada pela razão X/R (reatância indutiva pela resistência) do sistema (BALAMOUGAN; SIDHU, 2006).

O decaimento CC é um sinal não periódico e sua análise espectral revela as frequências no espectro, dado por $1/(a+j\omega)$, sendo “a” a própria razão X/R e ω a frequência angular em rad/s. A componente do decaimento CC afeta seriamente a precisão e a convergência dos algoritmos dos filtros digitais tais como os baseados em DFT, filtros cosseno, Kalman e least error square (LES). Logo, se no sinal de entrada está presente uma componente CC exponencial, esses algoritmos apresentam erros acentuados na estimação da componente fundamental (BENUMOUYAL, 1995; GIRGIS; BROWN, 1981). Assim técnicas especiais devem ser utilizadas para a extração da componente CC de modo a produzir uma estimação da componente fundamental precisa e rápida. O método apresentado neste artigo tem a vantagem de eliminar com eficiência grande parte da influência da componente CC e produzir estimações com precisão e velocidade.

Devido a especificações do método e características da tecnologia, foi escolhida para a realização deste trabalho plataformas Field Programmable Gate Arrays (FPGAs). Os FPGAs são circuitos integrados (CIs) que possuem blocos lógicos configuráveis interconectados entre si, podendo ser configurados para a realização de inúmeras tarefas. Realizam diversas funções de hardware sem a necessidade de se manusear fios nem materiais de solda. É desenvolvida uma aplicação em software e esta é compilada em um arquivo contendo as informações de como os blocos devem se arranjar dentro do chip para realizar a função de hardware desejada. Logo, basta fazer o download desse arquivo para o chip e o mesmo estará configurado da maneira que foi descrita no software.

Este trabalho utilizou um chip da família Stratix™, disponível em uma plataforma de desenvolvimento no laboratório. Essa família de FPGAs da ALTERA® é uma família de alto desempenho com densidade superior a 79040 elementos lógicos, e memória RAM superior a 7.5 Mbits. No mínimo 22 blocos DSP com 176 multiplicadores embarcados (9bits x 9bits) estão presentes em cada chip, o que torna o desenvolvimento de aplicações de processamento de sinais mais otimizadas

e facilita a implementação de filtros de alto desempenho. Oferece um sistema de gerenciamento de clock completo podendo atingir até 420 MHz, possuindo no mínimo 12 PLLs (Phase Locked Loops).

2 Metodologia

Dado um sinal de entrada da forma:

$$i[n] = I_0 e^{-\frac{nT_s}{\tau}} + \sum_{\substack{k=1 \\ k \text{ ímpar}}}^p I_k \text{sen}(k\omega_1 nT_s + \theta_k) \quad (1)$$

sendo I_0 a magnitude do decaimento CC exponencial, T_s o período de amostragem, τ a constante de tempo do decaimento CC exponencial, I_k e θ_k , magnitude e a fase do k -ésimo harmônico ímpar, ω_1 a frequência angular fundamental e p a ordem do maior harmônico ímpar presente no sinal.

A estimação do fasor da componente fundamental está ilustrada na Fig.1. Inicialmente o sinal é aplicado ao filtro DFT (MITRA, 2001). Paralelamente a este processo é realizado o cálculo das correções. O processo é finalizado quando é descontada da estimação do fasor da componente fundamental a contribuição do decaimento CC exponencial, gerando, desta maneira, a estimação do valor real do fasor da componente fundamental do sinal sem a influência do decaimento CC exponencial.

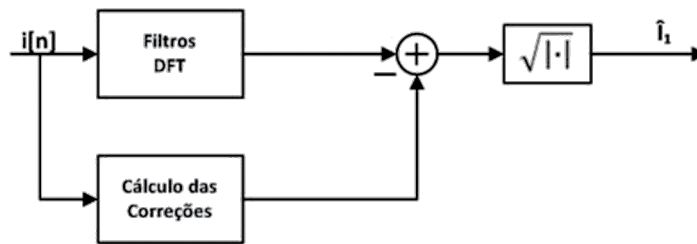


Fig. 1: Diagrama Global do Processo de Estimação

No processo de obtenção das correções, apresentado na Fig.2, o sinal é analisado utilizando-se uma janela adequada que contém zeros em locais estratégicos do espectro. Após o janelamento, é calculada a Discrete Time Fourier Transform (DTFT) do sinal janelado na frequência de 53 Hz (a escolha desta frequência será discutida posteriormente). A partir deste resultado o valor de $\tau(1)$ é estimado através da utilização da tabela k_1 e o valor de $I_0(1)$ é estimado através da utilização da tabela k_3 , que são criadas de forma off-line. Com os parâmetros do decaimento CC exponencial devidamente estimados e utilizando-se da tabela k_2 , também criada de forma off-line, é estimada então a contribuição do decaimento CC exponencial e, finalmente, geradas as correções.

Para a implementação em FPGA foi necessária a criação de um sinal de clock de baixa frequência, visto que a taxa de amostragem era muito baixa $64 \cdot 60 = 3840 \text{ Hz}$ e o gerador de clock disponível na placa de desenvolvimento era de 80 MHz, portanto, foram utilizados blocos PLL para a realização desta tarefa.

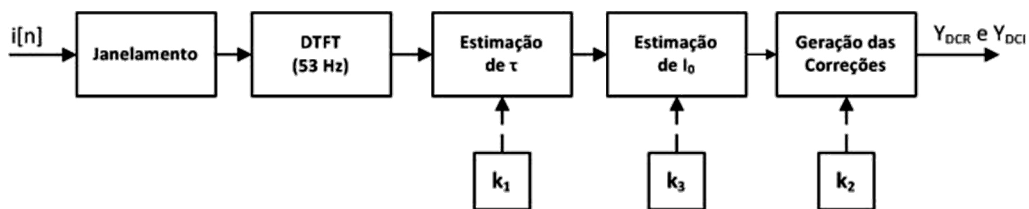


Fig. 2: Diagrama do Cálculo das Correções.

Filtros DFT

Filtros DFT são usados para estimar a amplitude da fundamental, neste trabalho foi feita a suposição que o sinal contenha somente harmônicos ímpares, porém a metodologia pode ser estendida para sinais que contenha também harmônicos pares, ao custo de um maior tempo de latência dos filtros. Com essas configurações, esse filtro é capaz de estimar a componente fundamental em 32 amostras do sinal, como mostra as equações (2) e (3).

$$\hat{Y}_R^{dft} = \frac{2}{N} \sum_{n=0}^{N-1} i[n] \cdot \cos\left(\frac{2\pi}{N}n\right) = Y_{CC_R} + Y_R^{f_1} \quad (2)$$

$$\hat{Y}_I^{dft} = \frac{2}{N} \sum_{n=0}^{N-1} i[n] \cdot \text{sen}\left(\frac{2\pi}{N}n\right) = Y_{CC_I} + Y_I^{f_1} \quad (3)$$

em que \hat{Y}_R^{dft} e \hat{Y}_I^{dft} são as partes real e imaginária, respectivamente, da estimação do fasor da componente fundamental, e N é o comprimento da janela utilizada. Como pode-se observar em (2) e (3) o resultado dessa estimação é composto pela soma de duas parcelas, uma relativa à componente fundamental, Y_{CC} , e outra à componente CC Y^f .

Para a implementação dos filtros DFT foi criada uma estrutura de multiplicação-acumulação, multiplexada, como mostrada na Fig.3. A estrutura foi criada de forma multiplexada, pois desta maneira economizou-se recursos de hardware, visto que o produto das posições da janela pelas posições do vetor de coeficientes não é feito paralelamente, em um único ciclo de clock, o que necessita de 36 multiplicadores. Da forma como foi implementado, o produto de cada posição da janela por cada coeficiente do filtro é realizado a cada ciclo de clock, sendo o resultado destes produtos acumulado no ACC.

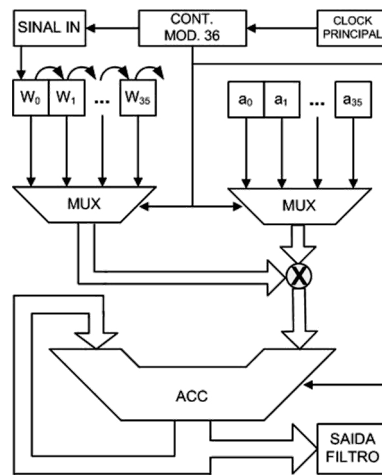


Fig. 3: Estrutura do Filtro Sintetizada no “FPGA”.

Janelamento do Sinal

O objetivo de janelar o sinal antes da aplicação da DTFT, apresentado em Mitra (2001), é aproveitar a propriedade da modulação, por $e^{-j\omega n}$ que esta, transformada, propicia. Desta forma, é possível deslocar a resposta em frequências da janela para a esquerda pelo fator ω , propiciando o posicionamento estratégico dos zeros da janela no espectro de frequências.

O processo utilizado para obtenção desta janela é a amostragem na frequência (MITRA, 2001). Neste processo os vetores de frequências positivas e negativas, utilizados para a obtenção dos zeros são:

$$f_n^+ = (n \cdot 60 + f') \quad (4)$$

$$f_n^- = (-n \cdot 60 + f') \quad (5)$$

em que $n=1,2,3,\dots,J-1$, onde J é o comprimento da janela desejada e f' a frequência em Hz na qual a DTFT será calculada. Adicionalmente, na frequência zero, a janela é forçada a ter magnitude unitária. Neste trabalho foi utilizado $J=36$ e $f'=53\text{Hz}$. A resposta em frequência da janela proposta é apresentada na Fig.4.

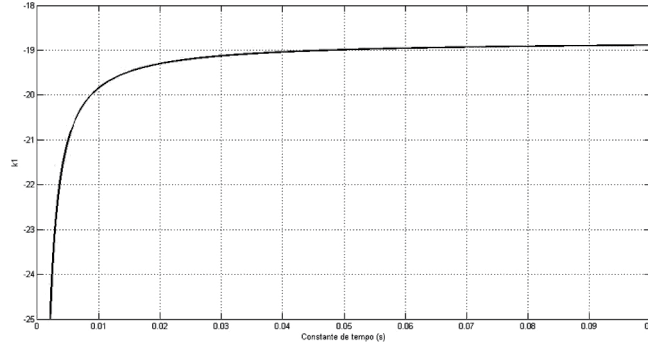


Fig. 4: Resposta em frequência da janela proposta.

Como pode ser observado em (4) e (5), os vetores de frequências são assimétricos, o que dará origem a um filtro, aqui denominado filtro DTFT, com resposta ao impulso contendo coeficientes complexos. Sejam os coeficientes do filtro dados por:

$$h[n] = h_R[n] + jh_I[n] \quad (6)$$

Aplicando a DTFT e a janela mostrada em (6) ao sinal de entrada mostrado em (1) temos:

$$Y(e^{j\omega'}) = \sum_{n=0}^{J-1} i[n] \cdot h[n] \cdot e^{-jn\omega'} = \sum_{n=0}^{J-1} I_0 e^{-\frac{nT_S}{\tau}} \cdot (a[n] + jb[n]) \quad (7)$$

em que $a[n]$ e $b[n]$ são os coeficientes do filtro DTFT depois de ser janelado. A partir de (7) serão estimados os parâmetros de τ e de I_0 .

Foi criada uma estrutura idêntica a da Fig.3 para a implementação deste filtro, porém foi necessária a criação de um artifício de sincronismo entre os dois filtros, visto que o filtro DFT trabalha com uma janela de dados de $N=32$, ao passo que o filtro DTFT janelado utiliza uma janela de $J=36$. Logo, incluiu-se mais 4 coeficientes nulos no filtro DFT, processo que não altera sua resposta em magnitude nem em fase, mas proporciona um atraso na resposta em quatro ciclos de clock. Com isso, os dois filtros geram suas respostas no mesmo instante de tempo.

Tabelas k_1 , k_2 e k_3

Foram criadas tabelas off-line visando reduzir o esforço computacional exigido pelo algoritmo, visto que o mesmo seria implementado em plataformas dedicadas e com operação em tempo real. Assim as operações que demandavam muito tempo de CPU, sempre quando possível, foram substituídas por buscas em tabelas.

Todas as tabelas são calculadas para valores de τ variando no intervalo de 0 ms a 100 ms, conforme Sidhu e outros (2005), mas com passo de 0,1 ms para aumentar a sensibilidade, evitando um processo de interpolação e, conseqüentemente, diminuindo o esforço computacional exigido pelo algoritmo.

A tabela k_1 foi criada relacionando valores de τ e de (8). A Fig.5 mostra esta relação, sendo a abscissa τ e a ordenada $k_1(\tau)$.

$$k_1(\tau) = \frac{\sum_{n=0}^{J-1} e^{-\frac{nT_S}{\tau}} \cdot b[n]}{\sum_{n=0}^{J-1} e^{-\frac{nT_S}{\tau}} \cdot a[n]} \quad (8)$$

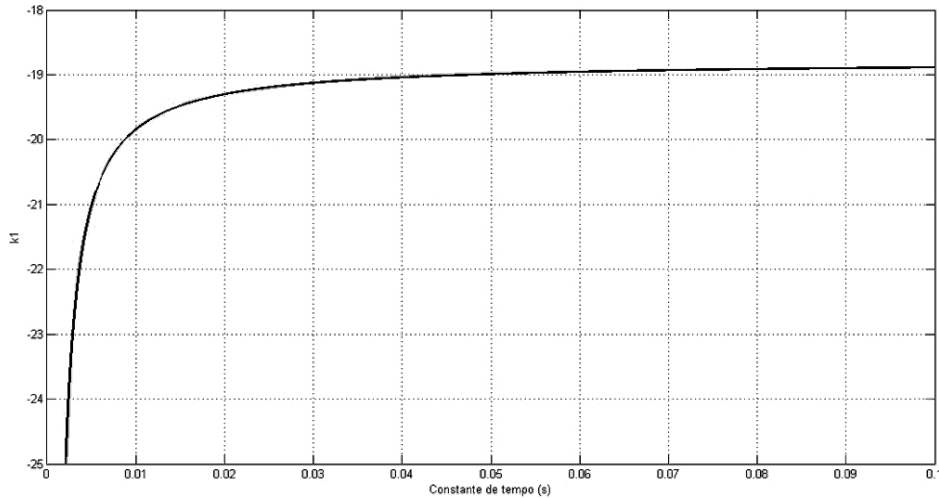


Fig. 5: Valores armazenados na tabela k_1 .

Observando-se a curva mostrada na Fig.5, nota-se que a partir de 0,03s a curva tende a um valor constante. Isto é ruim porque esta curva será discretizada, processo que tornará a tabela com muitos valores iguais, devido à precisão finita para a representação dos números, e armazenada em uma tabela, na qual posteriormente será realizada uma busca. Essa curva tem relação direta com o resultado do cálculo da DTFT, logo, variando-se a frequência na qual a DTFT é calculada varia-se também essa curva. Portanto escolheu-se calcular a DTFT na frequência que gerasse uma curva com maior variabilidade nesse intervalo, gerando assim, uma melhor representação quando a curva for discretizada. Após várias simulações, verificou-se que a frequência de 53 Hz era a que produzia a maior variabilidade na faixa em questão.

A tabela k_2 foi criada variando-se valores de τ na equação a seguir:

$$k_2(\tau) = \sum_{n=0}^{N-1} e^{-\frac{nT_S}{\tau}} \cdot e^{-j\left(\frac{2\pi f_1}{F_S}\right)n} \quad (9)$$

Os valores de $k_2(\tau)$ também são calculados para valores de τ variando no intervalo de 0 ms a 100 ms, com passo de 0,1 ms, sendo f_1 a frequência fundamental do sinal.

A tabela k_3 foi criada relacionando-se os valores de τ e da expressão a seguir:

$$k_3(\tau) = \sum_{n=0}^{N-1} \left\{ e^{-\frac{nT_S}{\tau}} \cdot a[n] \right\} \quad (10)$$

Esta tabela foi criada para os mesmos valores de τ que as tabelas anteriores e será utilizada para o cálculo da estimação de I_0 .

Todas as tabelas apresentadas anteriormente foram armazenadas em memórias ROM dentro do chip FPGA e durante a execução do algoritmo eram lidas em busca dos valores para a geração das correções a serem aplicadas na estimação realizada pelo filtro DFT.

Estimação de τ

O valor de τ é estimado via busca na tabela k_1 do seguinte valor (VIANELLO et al, 2009):

$$k_1(\hat{\tau}) = \frac{Y_I(e^{j\omega'})}{Y_R(e^{j\omega'})} \quad (11)$$

em que $Y_R(e^{j\omega'})$ e $Y_I(e^{j\omega'})$ são as partes real e imaginária de (7), e τ a estimativa de τ .

O processo de estimação de τ inicia-se assim que uma amostra é totalmente filtrada pelo filtro das correções. Então é realizada a divisão da parte real pela parte imaginária dessa resposta e o resultado é o valor a ser encontrado na tabela k_1 .

O processo de busca se baseia em uma busca pelo valor mais próximo ao valor da resposta do filtro. Quando este valor é encontrado, a posição dele na tabela tem uma relação direta com τ , sendo assim referência para a estimação de I_0 e para a geração das correções.

Foi adotado um algoritmo de busca binária como ferramenta para encontrar o valor de k_1 , devido a sua eficiência e rapidez. No caso de uma tabela com 996 elementos, ele é capaz de encontrar o valor mais próximo em apenas 11 iterações, ou seja, 11 ciclos de clock.

O algoritmo de busca binária pode ser visto como um jogo de adivinhação, no qual se tem uma lista ordenada e precisa-se determinar a posição desconhecida de um valor alvo (NOWAK, 2008). Esse processo é baseado em perguntas do tipo: “O valor alvo é maior ou menor do que $k_1(i)$?”, sendo $k_1(i)$ o valor procurado.

O passo inicial consiste em iniciar dois ponteiros, um ponteiro L na posição limite inferior da tabela, e outro H na posição limite superior. Feito isso, lê-se a posição $i = (L+H)/2$ e faz-se a pergunta: “O valor alvo é igual a $k_1(i)$?”. Se for, o processo é finalizado. Se o valor alvo for maior do que $k_1(i)$, seta-se o ponteiro $L = i$ e repete-se o processo. E se o valor alvo for menor do que $k_1(i)$, seta-se o ponteiro $H = i$ e repete-se o processo. A Figura 6: Esquema de busca binária em tabelas.

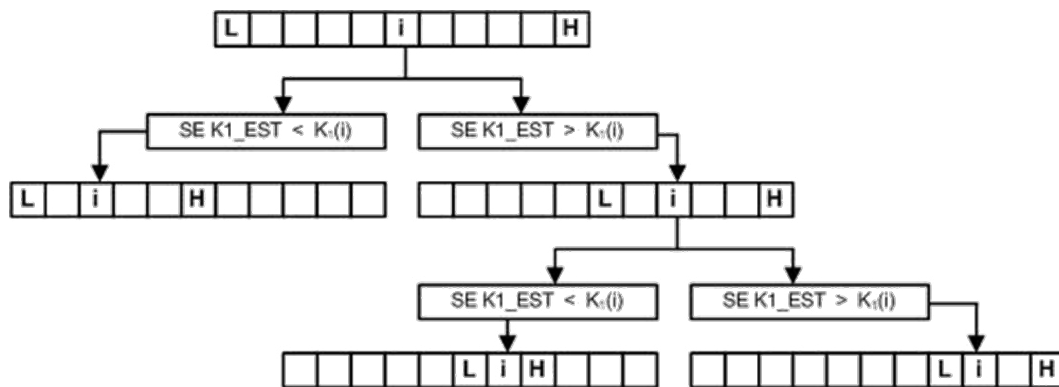


Fig. 6: ilustra o processo de busca binária.

Estimação de I_0

Partindo de (7) podemos estimar, com auxílio da tabela k_3 , o valor de I_0 da seguinte maneira:

$$\hat{I}_0 = \frac{Y_R(e^{j\omega'})}{\sum_{n=0}^{N-1} \left\{ e^{-\frac{nT_S}{\tau}} \cdot a[n] \right\}} = \frac{Y_R(e^{j\omega'})}{k_3(\hat{\tau})} \quad (12)$$

em que $Y_R(e^{j\omega'})$ é a parte real de (7), e I_0 a estimativa de I_0 .

A estimação de I_0 é feita com auxílio da tabela k_3 e do filtro das correções. Esta estimação é dada pela divisão da parte real do filtro de entrada pelo valor lido na posição referente à τ na tabela k_3 , se $\tau \neq 0$, ou pela soma dos coeficientes reais do filtro de correção, se $\tau = 0$, como mostrado na Figura 7.

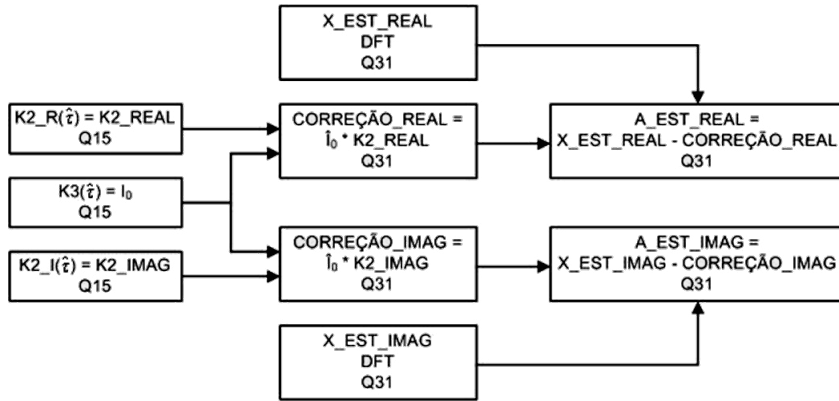


Fig. 7: Diagrama do cálculo da estimação de I_0 .

Geração das Correções

A geração das correções a serem feitas em (2) e (3) é realizada através de leitura na tabela k_2 . As correções são dadas por:

$$\hat{Y}_{DC} = \hat{I}_0 \cdot k_2(\hat{t}) = \hat{Y}_{DCR} - j\hat{Y}_{DCI} \quad (13)$$

em que Y_{DCR} e Y_{DCI} são as partes real e imaginária, respectivamente de Y_{DC} (estimação da influência do decaimento CC exponencial).

Pode-se então aplicar as correções obtidas (13) em (2) e (3), e se $Y_{DCR} = Y_{DCR}$ e $Y_{DCI} = Y_{DCI}$, então:

$$Y_R^{dft} = Y_{DCR} + Y_R^{f1} - \hat{Y}_{DCR} = Y_R^{f1} \quad (14)$$

$$Y_I^{dft} = Y_{DCI} + Y_I^{f1} + \hat{Y}_{DCI} = Y_I^{f1} \quad (15)$$

Obtêm-se, então, em (14) e (15), as estimações da componente fundamental livres da influência do decaimento CC.

Esse processo é baseado na estimação de I_0 e na posição de τ . Com a posição de τ encontra-se, através da leitura desse endereço, na tabela k_2 , o valor de k_2 , parte real e imaginária e é ilustrado na Fig.8.

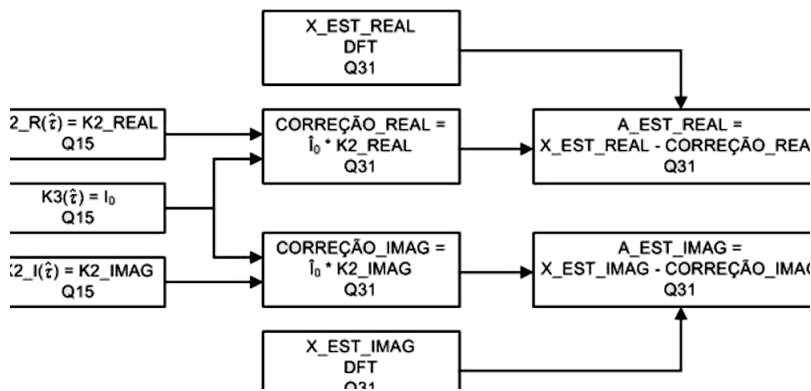


Fig. 8: Diagrama do cálculo da amplitude corrigida.

3 Resultados e discussões

Serão apresentados resultados de simulação funcional, on-line e com aritmética de ponto fixo, obtidos no software Modelsim®. Será feita uma comparação com os resultados obtidos no MATLAB,

de forma off-line e com aritmética de ponto flutuante, do algoritmo proposto e do algoritmo que utiliza filtros DFT, normalmente utilizados nos relés de proteção.

O caso aqui tratado, Fig.9, trata-se de uma onda que apresenta uma componente de forte decaimento CC exponencial, que é o foco do método proposto. Note que em menos de 1 ciclo o novo método estima o valor da componente fundamental, enquanto o método DFT só converge a um valor preciso após vários ciclos da fundamental.

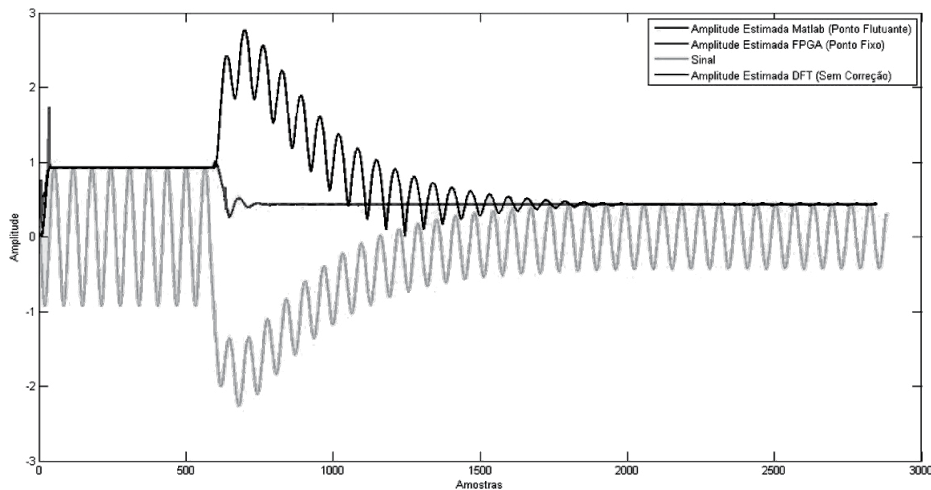


Figura 9: Resultado da estimação pelo método proposto

4 Conclusão

Este trabalho apresentou a implementação de um algoritmo de estimação de fasores na presença de decaimento CC exponencial em “Field Programmable Gate Array” (FPGA). A metodologia é baseada no uso de técnicas de processamento de sinais tais como janelamento, Discrete Time Fourier Transform (DTFT) e buscas em tabelas. O método é capaz de estimar a componente fundamental em aproximadamente meio ciclo da componente fundamental.

Apesar de estar em nível de simulação funcional, o código foi, em sua totalidade, escrito em linguagem Verilog sintetizável, não necessitando assim de modificações para ser transferido para o FPGA e sintetizar o hardware descrito.

Como pode-se observar na Fig. 9, apesar do resultado obtido em ponto fixo apresentar um pequeno erro em relação ao sinal obtido em ponto flutuante, apresenta uma melhora significativa em relação ao algoritmo normalmente utilizado nos relés de proteção.

FPGA IMPLEMENTATION OF A PHASOR ESTIMATOR IN PRESENCE OF DC EXPONENTIAL DECAYMENT COMPONENT

ABSTRACT

This paper presents the implementation in “Field Programmable Gate Array (FPGA) of a phasor estimation algorithm in the presence of exponential decayment. The proposed method is based on parameter estimation of exponential decayment and makes use of signal processing techniques such as windowing, “Discrete Time Fourier Transform (DTFT) and look up tables. The method is able to estimate the fundamental component in approximately half cycle of the fundamental component. Simulation results in real time using fixed-point arithmetic

are compared with those generated in floating point and off-line from MATLAB. Synthetic and real case examples were used to validate the estimation algorithm.

The estimation of the phasor of the fundamental component is performed by applying discrete Fourier filters (DFT). Corrections due to the presence of the exponential component are calculated parallel to this process and the final result is obtained when the corrections are applied to the estimation algorithm.

To obtain these corrections, the signal is analyzed using an appropriate window that contains zeros in strategic locations in the spectrum. Then, the DTFT of the resulting signal at a frequency of 53 Hz is calculated. From these results we estimate the parameters of the DC decayment, by look up tables. The tables are generated offline and stored in ROM memory instantiated in code. To simplify the implementation were used some “Megacores” provided by the manufacturer (Altera [®]), such as ROM memories and dividers. So the focus of this work was the synthesis filters and search algorithm in tables. Despite being at the level of simulation, the code was written in synthesizable language, so it is ready to be recorded on the chip.

Keywords: Estimation, Exponential DC decayment, Windowing, FPGA, Look-up-tables, Filters, DFT, DTFT, Corrections.

5 Agradecimentos

Os autores agradecem ao CNPq, à CAPES e à FAPEMIG que financiaram parcialmente este trabalho.

Referências

BALAMOUGAN, V.; SIDHU, T. S. “A New Filtering Technique to Eliminate Decaying CC and Harmonics for Power System Phasor Estimation” IEE, New York, n. 2006.

BENUMOUYAL, G. “Removal of CC-offset in current waveforms using digital mimic filtering” **IEEE Trans. Power Del.**, v. 10, no. 2, p. 621–630, Apr. 1995.

GIRGIS, A. A. ; BROWN, R. G. “Application of Kalman Filtering in Computer Relaying” **IEEE Trans. On Power Apparatus and Systems**, v.100, no.7, p.3387-3397, July 1981.

MITRA, S. K. **Digital Signal Processing - A Computer Based Approach**. 2nd Ed. New York: McGraw-Hill, 2001.

NOWAK, R. “Generalized Binary Search”, Communication, Control, and Computing, 2008 46th Annual Allerton Conference on 23-26 Sept. 2008, p.568 – 574.

SIDHU, T. S.; ZHANG X.; BALAMOUGAN, V. “A New Half-Cycle Phasor Estimation Algorithm” **IEEE Trans. Power Del.**, v.20, no.2, p.1299–1305, Apr. 2005.

VIANELLO, R.; MANSO, L.; DUQUE, C. A.; CERQUEIRA, A. S.; RIBEIRO, M, V.; SILVEIRA, P. M.; RIBEIRO, P. F.; “Nova Técnica de Estimação de Fasores na Presença de Decaimento CC Exponencial em Sistemas de Potencia”, **VIII CBQEE**, Blumenau-SC, Brasil, 2009.